



Khai phá dữ liệu

NHẬP MÔN

PGS. TS. Nguyễn Thanh Bình

AISIA Research Lab

Trường Đại học Khoa học Tự nhiên
Đại học Quốc gia Thành phố Hồ Chí Minh

Ngày 25 tháng 02 năm 2023

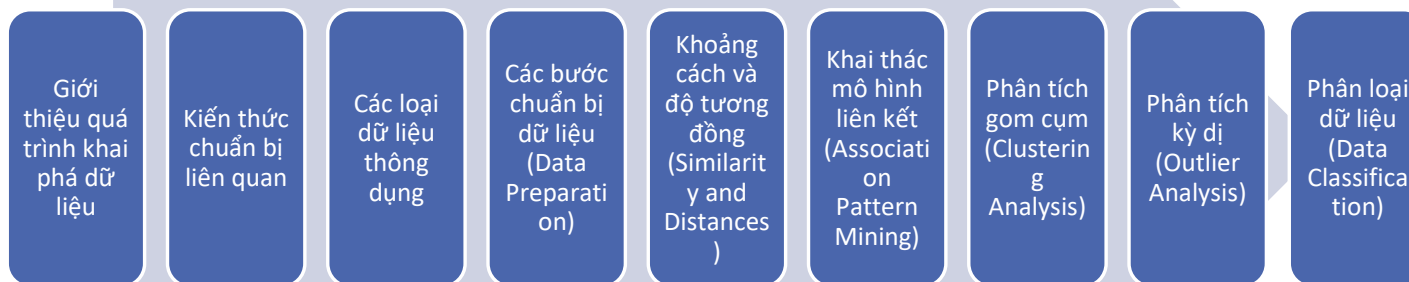


Hello!

Giảng viên: PGS.TS. **Nguyễn Thanh Bình**

- Tốt nghiệp Tiến Sĩ tại Đại Học Bách Khoa Paris năm 2012.
- Trưởng Bộ Môn Ứng Dụng Tin Học, trường Đại Học Khoa Học Tự Nhiên, Đại học Quốc Gia TP HCM.
- Phụ trách chuyên môn chương trình Thạc Sĩ chuyên ngành Khoa Học Dữ Liệu tại trường Đại Học Khoa Học Tự Nhiên, Đại học Quốc Gia TP HCM.
<https://www.facebook.com/master.datascience.hcmus>
- Có hơn 65 bài báo nghiên cứu khoa học với một số công trình được báo cáo tại các hội nghị uy tín: NeurIPS, WACV, PAKDD, ACM ICMR, MobiSys, và các tạp chí khoa học uy tín.
- Quả cầu vàng Khoa Học Công Nghệ Việt Nam năm 2021.

Nội dung môn học



Hình thức học - thi

Thời gian học: E202B, thứ 7 hàng tuần từ 15h đến 17h30.

Địa điểm: trường Đại học Khoa học Tự nhiên Tp Hồ Chí Minh

Nội quy: đi học đúng giờ

Hình thức kiểm tra và thi: 30% (bài tập trên lớp) + 70% đồ án cuối môn học. Mỗi học viên sẽ có một đồ án riêng.

Email: ngtbinh@hcmus.edu.vn

Khi nộp bài tập trên lớp, các bạn nộp thông qua email với tiêu đề có format: [K31-KHDL-Mã số học viên] < Nội dung >

Giáo trình môn học: Data Mining (Charu C. Aggarwal)

Lời mở đầu

Khai thác dữ liệu là ngành nghiên cứu khoa học liên quan đến việc

- thu thập
- làm sạch
- xử lý
- phân tích
- đạt được những thông tin hữu ích từ dữ liệu thu thập.

Có rất nhiều biến thể liên quan đến khai thác dữ liệu tùy thuộc vào

- dạng bài toán
- các ứng dụng
- công thức
- cách thức biểu diễn dữ liệu gặp phải trong các ứng dụng thực tế.

Lời mở đầu

Do đó, việc “khai thác dữ liệu” là một thuật ngữ có tính phổ quát cao và được sử dụng để mô tả các khía cạnh khác nhau của quá trình xử lý dữ liệu.



Lời mở đầu

Ngày nay, dữ liệu thu thập được là vô cùng lớn từ các hệ thống ở các công ty và các tập đoàn trên thế giới.

World Wide Web: The number of documents on the indexed Web is now on the order of billions, and the invisible Web is much larger. User accesses to such documents create Web access logs at servers and customer behavior profiles at commercial sites. Furthermore, the linked structure of the Web is referred to as the *Web graph*, which is itself a kind of data. These different types of data are useful in various applications. For example, the Web documents and link structure can be mined to determine associations between different topics on the Web. On the other hand, user access logs can be mined to determine frequent patterns of accesses or unusual patterns of possibly unwarranted behavior.

Lời mở đầu

Financial interactions: Most common transactions of everyday life, such as using an automated teller machine (ATM) card or a credit card, can create data in an automated way. Such transactions can be mined for many useful insights such as fraud or other unusual activity.

User interactions: Many forms of user interactions create large volumes of data. For example, the use of a telephone typically creates a record at the telecommunication company with details about the duration and destination of the call. Many phone companies routinely analyze such data to determine relevant patterns of behavior that can be used to make decisions about network capacity, promotions, pricing, or customer targeting.

Sensor technologies and the Internet of Things: A recent trend is the development of low-cost wearable sensors, smartphones, and other smart devices that can communicate with one another. By one estimate, the number of such devices exceeded the number of people on the planet in 2008 [30]. The implications of such massive data collection are significant for mining algorithms.

Lời mở đầu

DATA MINING



Dữ liệu thô từ các nguồn dữ liệu khác nhau có thể là các format tùy ý, không có cấu trúc hoặc có cấu trúc. Thậm chí, có một số trường hợp dữ liệu thô có dạng bán cấu trúc hoặc thậm chí ở định dạng không phù hợp ngay lập tức để có thể xử lý tự động hiệu quả.

Với nguồn dữ liệu dồi dào từ khách hàng và các hệ thống/sản phẩm của công ty, nhu cầu xây dựng các hệ thống phân tích và mô tả dữ liệu là **vô cùng to lớn**.

Lời mở đầu



Ví dụ, dữ liệu được thu thập thủ công có thể được lấy từ các nguồn không đồng nhất ở các định dạng khác nhau và nhưng bằng cách nào đó cần phải được xử lý bởi một chương trình máy tính tự động để có được thông tin chi tiết.

Các nhà phân tích khai thác dữ liệu sẽ sử dụng một hệ thống xử lý, trong đó, các nguyên dữ liệu đầu vào sẽ được:

- thu thập
- làm sạch
- chuyển đổi thành một định dạng tiêu chuẩn hóa.

Lời mở đầu



Dữ liệu có thể có nhiều loại định dạng hoặc các kiểu khác nhau. Các loại dữ liệu ở đây có thể là

- định lượng (ví dụ: tuổi),
- phân loại (ví dụ: giới tính),
- văn bản,
- không gian, thời gian
- biểu đồ.

Lời mở đầu



- Mặc dù cho đến nay phần lớn các dạng dữ liệu phổ biến là đa chiều, các loại dữ liệu có cấu trúc phức tạp hơn dần dần có tỷ lệ ngày càng tăng so với trước đây.

Quá trình khai phá dữ liệu

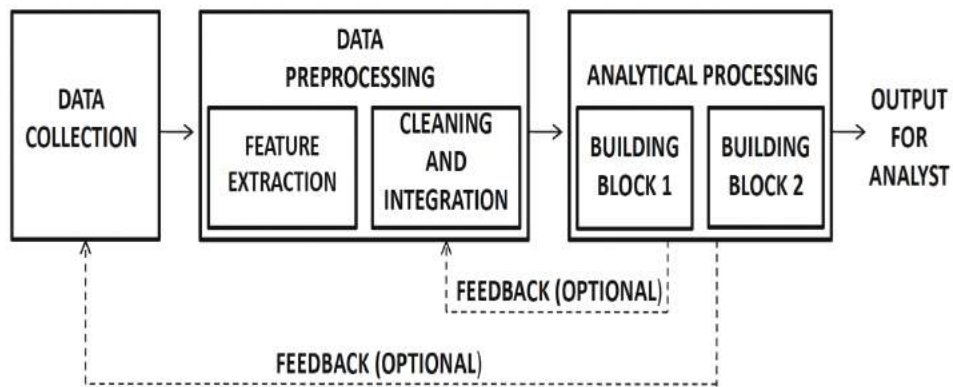


Figure 1.1: The data processing pipeline

Quá trình khai thác dữ liệu là một quá trình bao gồm nhiều giai đoạn, chẳng hạn như làm sạch dữ liệu, trích xuất tính năng, và thiết kế thuật toán.



Đặc điểm chung của quá trình khai phá dữ liệu



Quá trình khai phá dữ liệu

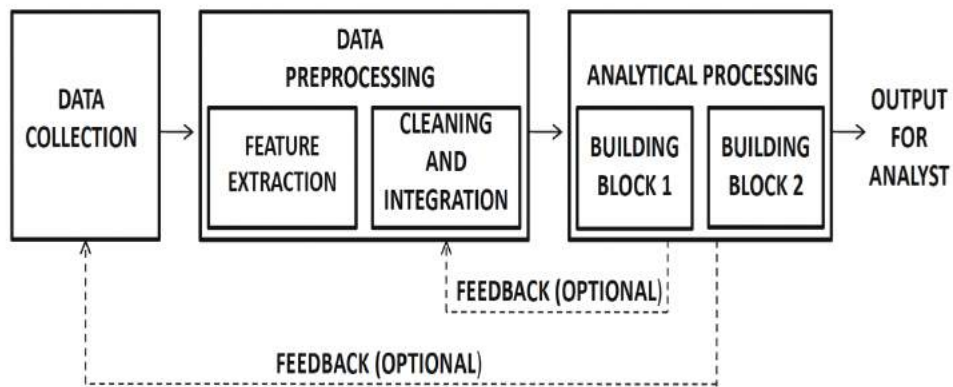


Figure 1.1: The data processing pipeline

Quá trình khai thác dữ liệu là một quá trình bao gồm nhiều giai đoạn, chẳng hạn như làm sạch dữ liệu, trích xuất tính năng, và thiết kế thuật toán.

Giai đoạn tiền xử lý dữ liệu

- Đây là thể là giai đoạn quan trọng nhất trong quá trình xử lý dữ liệu.
- Giai đoạn này nối tiếp sau việc thu thập dữ liệu và có thể được chia thành các bước sau.
 - Trích xuất tính năng
 - Làm sạch dữ liệu
 - Chọn lọc tính năng và biến đổi

Giai đoạn tiền xử lý dữ liệu

- Trích xuất tính năng

Feature extraction: An analyst may be confronted with vast volumes of raw documents, system logs, or commercial transactions with little guidance on how these raw data should be transformed into meaningful database features for processing. This phase is highly dependent on the analyst to be able to abstract out the features that are most relevant to a particular application. For example, in a credit-card fraud detection application, the amount of a charge, the repeat frequency, and the location are often good indicators of fraud. However, many other features may be poorer indicators of fraud. Therefore, extracting the right features is often a skill that requires an understanding of the specific application domain at hand.

Source: Data Mining, Charu C. Aggarwal

Giai đoạn tiền xử lý dữ liệu

- Làm sạch dữ liệu

Data cleaning: The extracted data may have erroneous or missing entries. Therefore, some records may need to be dropped, or missing entries may need to be estimated. Inconsistencies may need to be removed.

Giai đoạn tiền xử lý dữ liệu

- Chọn lọc tính năng và biến đổi

Feature selection and transformation: When the data are very high dimensional, many data mining algorithms do not work effectively. Furthermore, many of the high-dimensional features are noisy and may add errors to the data mining process. Therefore, a variety of methods are used to either remove irrelevant features or transform the current set of features to a new data space that is more amenable for analysis. Another related aspect is data *transformation*, where a data set with a particular set of attributes may be transformed into a data set with another set of attributes of the same or a different type. For example, an attribute, such as age, may be partitioned into ranges to create discrete values for analytical convenience.

Source: Data Mining, Charu C. Aggarwal

Giai đoạn phân tích

Việc mỗi ứng dụng khai phá dữ liệu đều khác nhau khiến việc tạo các kỹ thuật tổng quát có thể tái sử dụng rất khó.

Tuy nhiên, vẫn có nhiều thiết lập khai phá dữ liệu có thể được tái sử dụng với các hoàn cảnh ứng dụng khác nhau.

Việc sử dụng này phụ thuộc vào kỹ năng và kinh nghiệm.

Các loại dữ liệu cơ bản

Có 2 loại chính với độ phức tạp đa dạng cho quá trình khai phá dữ liệu.

- Dữ liệu định hướng không phụ thuộc.
- Dữ liệu định hướng phụ thuộc.

Các loại dữ liệu cơ bản

- Dữ liệu định hướng **không** phụ thuộc.

Nondependency-oriented data: This typically refers to simple data types such as multi-dimensional data or text data. These data types are the simplest and most commonly encountered. In these cases, the data records do not have any specified dependencies between either the data items or the attributes. An example is a set of demographic records about individuals containing their age, gender, and ZIP code.

Dependency-oriented data: In these cases, implicit or explicit relationships may exist between data items. For example, a social network data set contains a set of *vertices* (data items) that are connected together by a set of *edges* (relationships). On the other hand, time series contains implicit dependencies. For example, two successive values collected from a sensor are likely to be related to one another. Therefore, the time attribute implicitly specifies a dependency between successive readings.

Dữ liệu định hướng **không** phụ thuộc

Là dạng dữ liệu đơn giản nhất và thường được gọi là dữ liệu nhiều chiều.

Table 1.1: An example of a multidimensional data set

Name	Age	Gender	Race	ZIP code
John S.	45	M	African American	05139
Manyona L.	31	F	Native American	10598
Sayani A.	11	F	East Indian	10547
Jack M.	56	M	Caucasian	10562
Wei L.	63	M	Asian	90210


Definition 1.3.1 (Multidimensional Data) A multidimensional data set \mathcal{D} is a set of n records, $\overline{X}_1 \dots \overline{X}_n$, such that each record \overline{X}_i contains a set of d features denoted by $(x_i^1 \dots x_i^d)$.



Dữ liệu định hướng **không** phụ thuộc

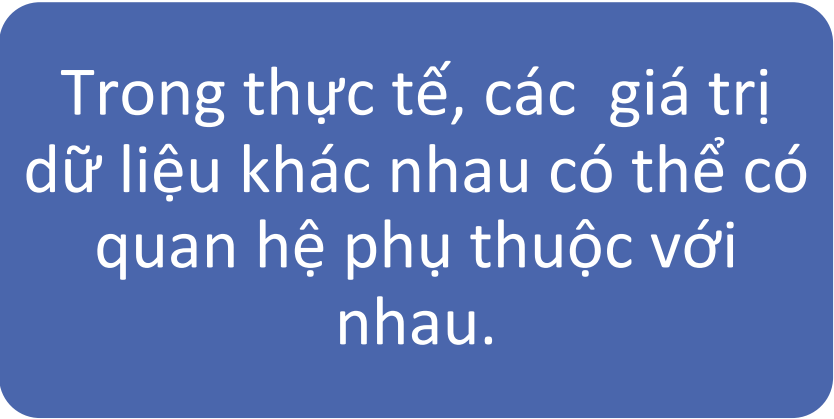


Dữ liệu nhiều chiều còn có thể được phân thành các loại.

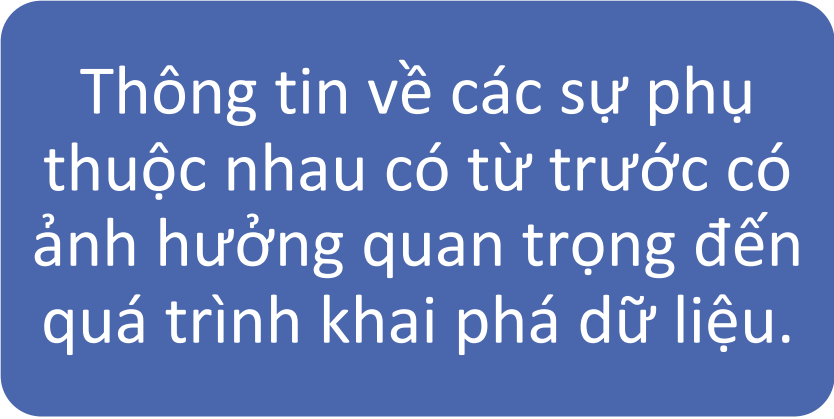
- Dữ liệu nhiều chiều định lượng được.
 - Dữ liệu phân loại và dữ liệu thuộc tính trộn.
 - Dữ liệu nhị phân và dữ liệu tập hợp.
 - Dữ liệu văn bản.
- 




Dữ liệu định hướng phụ thuộc



Trong thực tế, các giá trị dữ liệu khác nhau có thể có quan hệ phụ thuộc với nhau.



Thông tin về các sự phụ thuộc nhau có từ trước có ảnh hưởng quan trọng đến quá trình khai phá dữ liệu.






Dữ liệu định hướng phụ thuộc



Các loại phụ thuộc trong dữ liệu có thể được phân thành 2 loại sau.

- Phụ thuộc ngầm.
 - Phụ thuộc tường minh.
- 



Dữ liệu định hướng phụ thuộc

Phụ thuộc ngầm.

Implicit dependencies: In this case, the dependencies between data items are not explicitly specified but are known to “typically” exist in that domain. For example, consecutive temperature values collected by a sensor are likely to be extremely similar to one another. Therefore, if the temperature value recorded by a sensor at a particular time is significantly different from that recorded at the next time instant then this is extremely unusual and may be interesting for the data mining process. This is different from multidimensional data sets where each data record is treated as an independent entity.



Dữ liệu định hướng phụ thuộc

Phụ thuộc tường minh.



Explicit dependencies: This typically refers to graph or network data in which edges are used to specify explicit relationships. Graphs are a very powerful abstraction that are often used as an intermediate representation to solve data mining problems in the context of other data types.



Dữ liệu định hướng phụ thuộc



Dữ liệu định hướng phụ thuộc có thể được phân thành các loại sau.

- Dữ liệu chuỗi thời gian (time-series).
 - Dãy và chuỗi (strings) rời rạc.
 - Dữ liệu không gian.
 - Dữ liệu mạng và đồ thị.
- 
- 



Dữ liệu định hướng phụ thuộc

Dữ liệu chuỗi thời gian (time-series)

Definition 1.3.2 (Multivariate Time-Series Data) *A time series of length n and dimensionality d contains d numeric features at each of n time stamps $t_1 \dots t_n$. Each time-stamp contains a component for each of the d series. Therefore, the set of values received at time stamp t_i is $\bar{Y}_i = (y_i^1 \dots y_i^d)$. The value of the j th series at time stamp t_i is y_i^j .*



Dữ liệu định hướng phụ thuộc

Dãy và chuỗi (strings) rời rạc

Definition 1.3.3 (Multivariate Discrete Sequence Data) *A discrete sequence of length n and dimensionality d contains d discrete feature values at each of n different time stamps $t_1 \dots t_n$. Each of the n components \overline{Y}_i contains d discrete behavioral attributes $(y_i^1 \dots y_i^d)$, collected at the i th time-stamp.*



Dữ liệu định hướng phụ thuộc

Dữ liệu không gian

Definition 1.3.4 (Spatial Data) *A d -dimensional spatial data record contains d behavioral attributes and one or more contextual attributes containing the spatial location. Therefore, a d -dimensional spatial data set is a set of d dimensional records $\overline{X}_1 \dots \overline{X}_n$, together with a set of n locations $L_1 \dots L_n$, such that the record \overline{X}_i is associated with the location L_i .*



Dữ liệu định hướng phụ thuộc

Dữ liệu mạng và đồ thị

Definition 1.3.5 (Network Data) *A network $G = (N, A)$ contains a set of nodes N and a set of edges A , where the edges in A represent the relationships between the nodes. In some cases, an attribute set \overline{X}_i may be associated with node i , or an attribute set \overline{Y}_{ij} may be associated with edge (i, j) .*

Các khối xây dựng chính

Có 4 bài toán nền tảng trong quá trình khai phá dữ liệu.

Khai phá mẫu liên hệ.

Gom cụm dữ liệu.

Phát hiện ngoại lai.

Phân loại dữ liệu.

Các khối xây dựng chính

Khai phá mẫu liên hệ

Dạng sơ khởi nhất của bài toán này được định nghĩa với các cơ sở dữ liệu nhị phân thưa.

Definition 1.4.1 (Frequent Pattern Mining) *Given a binary $n \times d$ data matrix D , determine all subsets of columns such that all the values in these columns take on the value of 1 for at least a fraction s of the rows in the matrix. The relative frequency of a pattern is referred to as its support. The fraction s is referred to as the minimum support.*

Các khối xây dựng chính

Khai phá mẫu liên hệ

Ban đầu được đề xuất với bối cảnh khai phá quy luật liên hệ, với thêm một bước dựa trên một độ đo là “độ tin cậy” của quy luật.

Definition 1.4.2 (Association Rules) *Let A and B be two sets of items. The rule $A \Rightarrow B$ is said to be valid at support level s and confidence level c , if the following two conditions are satisfied:*

- 1. The support of the item set A is at least s .*
- 2. The confidence of $A \Rightarrow B$ is at least c .*

Các khối xây dựng chính

Gom cụm dữ liệu

Definition 1.4.3 (Data Clustering) *Given a data matrix D (database \mathcal{D}), partition its rows (records) into sets $\mathcal{C}_1 \dots \mathcal{C}_k$, such that the rows (records) in each cluster are “similar” to one another.*

Các khối xây dựng chính

Gom cụm dữ liệu

- Có một số ứng dụng quan trọng như sau.
 - Chia nhóm khách hàng.
 - Tóm tắt dữ liệu.
 - Ứng dụng vào các bài toán khai phá dữ liệu khác.

Các khối xây dựng chính

Phát hiện ngoại lai

Definition 1.4.4 (Outlier Detection) *Given a data matrix D , determine the rows of the data matrix that are very different from the remaining rows in the matrix.*

Các khối xây dựng chính

Phát hiện ngoại lai

Một số ứng dụng
quan trọng như sau.

Phát hiện xâm nhập.

Phát hiện gian lận
thẻ tín dụng.

Phát hiện các sự kiện
đáng quan tâm từ
thông tin sensor.

Chẩn đoán y khoa.

Các khối xây dựng chính

Phân loại dữ liệu

Definition 1.4.5 (Data Classification) *Given an $n \times d$ training data matrix D (database \mathcal{D}), and a class label value in $\{1 \dots k\}$ associated with each of the n rows in D (records in \mathcal{D}), create a training model \mathcal{M} , which can be used to predict the class label of a d -dimensional record $\bar{Y} \notin \mathcal{D}$.*

Các khối xây dựng chính

Phân loại dữ liệu có một số ứng dụng quan trọng như sau.

- Marketing có mục tiêu.
- Phát hiện xâm nhập.
- Phát hiện bất thường có giám sát.

Các khối xây dựng chính

- Các loại dữ liệu cụ thể có ảnh hưởng lớn đến loại bài toán có thể được đặt ra, đặc biệt là các loại dữ liệu phức tạp.

Table 1.2: Some examples of variation in problem definition with data type

Problem	Time series	Spatial	Sequence	Networks
Patterns	Motif-mining Periodic pattern	Colocation patterns	Sequential patterns Periodic Sequence	Structural patterns
	Trajectory patterns			
Clustering	Shape clusters	Spatial clusters	Sequence clusters	Community detection
	Trajectory clusters			
Outliers	Position outlier Shape outlier	Position outlier Shape outlier	Position outlier Combination outlier	Node outlier Linkage outlier Community outliers
	Trajectory outliers			
Classification	Position classification Shape classification	Position classification Shape classification	Position classification Sequence classification	Collective classification Graph classification
	Trajectory classification			

Các vấn đề về khả năng mở rộng và tình huống dòng dữ liệu

Với lượng dữ liệu ngày càng lớn thì chúng ta có 2 tình huống mở rộng như sau.

- Dữ liệu được chứa trên một hoặc nhiều máy, nhưng quá nhiều để xử lý một cách hiệu quả.
- Dữ liệu được sinh ra liên tục theo thời gian và với lượng lớn, việc lưu trữ toàn bộ không thực tế. Đây là tình huống dòng dữ liệu.

Các vấn đề về khả năng mở rộng và tình huống dòng dữ liệu

- Tình huống dòng dữ liệu có các thử thách lớn như sau.
 1. *One-pass constraint*: The algorithm needs to process the entire data set in one pass. In other words, after a data item has been processed and the relevant summary insights have been gleaned, the raw item is discarded and is no longer available for processing. The amount of data that may be processed at a given time depends on the storage available for retaining segments of the data.
 2. *Concept drift*: In most applications, the data distribution changes over time. For example, the pattern of sales in a given hour of a day may not be similar to that at another hour of the day. This leads to changes in the output of the mining algorithms as well.

Một số tình huống ứng dụng

- Xác định vị trí để đặt sản phẩm trong cửa hàng.

Application 1.6.1 (Store Product Placement) *A merchant has a set of d products together with previous transactions from the customers containing baskets of items bought together. The merchant would like to know how to place the product on the shelves to increase the likelihood that items that are frequently bought together are placed on adjacent shelves.*

- Gợi ý sản phẩm cho khách hàng.

Application 1.6.2 (Product Recommendations) *A merchant has an $n \times d$ binary matrix D representing the buying behavior of n customers across d items. It is assumed that the matrix is sparse, and therefore each customer may have bought only a few items. It is desirable to use the product associations to make recommendations to customers.*

Một số tình huống ứng dụng

- Xác định bất thường trong log lưu trữ của trang web.

Application 1.6.4 (Web Log Anomalies) *A set of Web logs is available. It is desired to determine the anomalous sequences from the Web logs.*

- Chẩn đoán y khoa.

Application 1.6.3 (Medical ECG Diagnosis) *Consider a set of ECG time series that are collected from different patients. It is desirable to determine the anomalous series from this set.*



Chuẩn bị dữ liệu



Chuẩn bị dữ liệu

- Định dạng của dữ liệu thực tế rất đa dạng.
 - Trong dữ liệu có thể có nhiều giá trị bị thiếu, không đồng nhất hoặc có lỗi sai.
- > Có nhiều thử thách khi muốn sử dụng dữ liệu hiệu quả.

Chuẩn bị dữ liệu

- Giai đoạn chuẩn bị dữ liệu là một quá trình nhiều tầng bao gồm nhiều bước riêng biệt.
- Tùy thuộc mỗi ứng dụng cụ thể thì một số hoặc tất cả bước này được sử dụng.

Chuẩn bị dữ liệu

Các bước này bao gồm

- Trích xuất đặc trưng và khả năng biến đổi kiểu dữ liệu.
- Làm sạch dữ liệu.
- Rút gọn, chọn lọc và biến đổi dữ liệu.

Chuẩn bị dữ liệu

Trích xuất đặc trưng và khả năng biến đổi của kiểu dữ liệu.

Dữ liệu thô thường không thích hợp để xử lý.

Chúng ta muốn lấy ra các đặc trưng có ý nghĩa từ dữ liệu.

Các đặc trưng khả diễn giúp việc hiểu các kết quả trung gian đơn giản hơn.

Chuẩn bị dữ liệu

Trích xuất đặc trưng và khả năng biến đổi của dạng dữ liệu.

Trong một số trường hợp dữ liệu được lấy từ nhiều nguồn và cần tích hợp lại vào cùng một cơ sở dữ liệu để xử lý.

Một số thuật toán chỉ hoạt động với một dạng dữ liệu cụ thể trong khi dữ liệu có thể có các kiểu dữ liệu không thuần nhất.

=> Khả năng biến đổi kiểu dữ liệu: khả năng từ một kiểu dữ liệu biến đổi sang kiểu khác có thể giúp xây dựng một bộ dữ liệu thuần nhất hơn cho các thuật toán xử lý.



Chuẩn bị dữ liệu

Làm sạch dữ liệu

Các nhập liệu bị thiếu, sai hoặc không đồng nhất được loại bỏ.

Bên cạnh đó, một số nhập liệu bị thiếu cũng có thể được ước lượng bằng các phương pháp điền khuyết.

Chuẩn bị dữ liệu

Rút gọn chọn lọc và biến đổi dữ liệu

Kích thước dữ liệu được giảm thông qua việc chọn lọc dữ liệu, chọn lọc đặc trưng hoặc biến đổi dữ liệu.

Kích thước dữ liệu giảm thường giúp các thuật toán chạy hiệu quả hơn.

Nếu các đặc trưng hoặc bản ghi không quan trọng bị loại bỏ thì chất lượng quá trình khai phá dữ liệu cũng được cải thiện.

Chuẩn bị dữ liệu

Các bước này bao gồm

- **Trích xuất đặc trưng** và khả năng biến đổi kiểu dữ liệu.
- Làm sạch dữ liệu.
- Rút gọn, chọn lọc và biến đổi dữ liệu.

Trích xuất đặc trưng

Trích xuất đặc trưng rất phụ thuộc vào từng ứng dụng cụ thể và bản nguồn lấy dữ liệu.

- Dữ liệu sensor.
- Dữ liệu ảnh.
- Log các trang web.
- Lưu thông mạng.
- Dữ liệu văn bản.

Trích xuất đặc trưng

- Trong một số trường hợp, trích xuất đặc trưng có quan hệ mật thiết với khái niệm về khả năng biến đổi kiểu dữ liệu.
- Với khả năng biến đổi kiểu dữ liệu, các đặc trưng bậc thấp của một kiểu dữ liệu có thể được biến đổi thành các đặc trưng bậc cao hơn của một kiểu dữ liệu khác.

Chuẩn bị dữ liệu

Các bước này bao gồm

- Trích xuất đặc trưng và **khả năng biến đổi kiểu dữ liệu.**
- Làm sạch dữ liệu.
- Rút gọn, chọn lọc và biến đổi dữ liệu.

Khả năng biến đổi kiểu dữ liệu

- Khả năng biến đổi kiểu dữ liệu có vai trò rất quan trọng do dữ liệu thường không thuần nhất mà chứa nhiều kiểu khác nhau.
- Với các dữ liệu không thuần nhất như vậy, chúng ta sẽ có các vấn đề sau.
 - Cần thiết kế thuật toán cho một tổ hợp kiểu dữ liệu bất kì.
 - Khó sử dụng các công cụ xử lý có sẵn.
- Trong một số trường hợp, việc biến đổi kiểu dữ liệu dẫn đến việc mất độ chính xác và tính biểu đạt.

Khả năng biến đổi kiểu dữ liệu

- Chúng ta có một số phép biến đổi giữa các kiểu dữ liệu như sau.

Table 2.1: Portability of different data types

Source data type	Destination data type	Methods
Numeric	Categorical	Discretization
Categorical	Numeric	Binarization
Text	Numeric	Latent semantic analysis (<i>LSA</i>)
Time series	Discrete sequence	<i>SAX</i>
Time series	Numeric multidimensional	<i>DWT, DFT</i>
Discrete sequence	Numeric multidimensional	<i>DWT, DFT</i>
Spatial	Numeric multidimensional	2-d <i>DWT</i>
Graphs	Numeric multidimensional	<i>MDS</i> , spectral
Any type	Graphs	Similarity graph (Restricted applicability)

Khả năng biến đổi kiểu dữ liệu

- Biến đổi dữ liệu số thành dữ liệu phân loại: rời rạc hóa.
- Biến đổi dữ liệu phân loại thành dữ liệu số: nhị phân hóa.
- Biến đổi dữ liệu văn bản thành dữ liệu số.
- Biến đổi dữ liệu chuỗi thời gian thành dữ liệu dãy rời rạc.
- Biến đổi dữ liệu chuỗi thời gian thành dữ liệu số.

Table 2.1: Portability of different data types

Source data type	Destination data type	Methods
Numeric	Categorical	Discretization
Categorical	Numeric	Binarization
Text	Numeric	Latent semantic analysis (<i>LSA</i>)
Time series	Discrete sequence	<i>SAX</i>
Time series	Numeric multidimensional	<i>DWT, DFT</i>
Discrete sequence	Numeric multidimensional	<i>DWT, DFT</i>
Spatial	Numeric multidimensional	2-d <i>DWT</i>
Graphs	Numeric multidimensional	<i>MDS</i> , spectral
Any type	Graphs	Similarity graph (Restricted applicability)

Source: Data Mining, Charu C. Aggarwal

Khả năng biến đổi kiểu dữ liệu

- Biến đổi dữ liệu dãy rời rạc thành dữ liệu số.
- Biến đổi dữ liệu không gian thành dữ liệu số.
- Biến đổi dữ liệu đồ thị thành dữ liệu số.
- Biến đổi dữ liệu bất kì thành dữ liệu đồ thị cho các ứng dụng về sự tương đồng.

Table 2.1: Portability of different data types

Source data type	Destination data type	Methods
Numeric	Categorical	Discretization
Categorical	Numeric	Binarization
Text	Numeric	Latent semantic analysis (<i>LSA</i>)
Time series	Discrete sequence	<i>SAX</i>
Time series	Numeric multidimensional	<i>DWT, DFT</i>
Discrete sequence	Numeric multidimensional	<i>DWT, DFT</i>
Spatial	Numeric multidimensional	2-d <i>DWT</i>
Graphs	Numeric multidimensional	<i>MDS</i> , spectral
Any type	Graphs	Similarity graph (Restricted applicability)

Source: Data Mining, Charu C. Aggarwal

Khả năng biến đổi kiểu dữ liệu

Biến đổi dữ liệu số thành dữ liệu phân loại: rời rạc hóa.

- Quá trình rời rạc hóa chia miền giá trị của thuộc tính số thành một số các khoảng.
- Khi đó, mỗi khoảng này có thể được gán một giá trị phân loại.
- Một khó khăn của việc này là dữ liệu có thể có phân phối không đều qua các khoảng.
- Tùy thuộc vào từng ứng dụng mà chúng ta có các cách rời rạc hóa khác nhau. Một số thí dụ về cách chọn khoảng khi rời rạc hóa như
 - Dùng các khoảng equi-width.
 - Dùng các khoảng equi-log.
 - Dùng các khoảng equi-depth.

Biến đổi dữ liệu phân loại thành dữ liệu số: nhị phân hóa.

- Có một số trường hợp chúng ta muốn sử dụng các thuật toán khai phá dữ liệu kiểu số với các dữ liệu kiểu phân loại.
- Với mỗi giá trị phân loại của thuộc tính phân loại, chúng ta sẽ dùng một thuộc tính nhị phân để biểu diễn.
- Với một giá trị phân loại, thuộc tính nhị phân tương ứng sẽ được gán giá trị 1 còn các thuộc tính nhị phân khác được gán giá trị 0.

Biến đổi dữ liệu văn bản thành dữ liệu số.

- Mặc dù dữ liệu văn bản có thể được biểu diễn theo kiểu số bằng các vector thưa với số chiều rất cao, cách này không thích hợp với các thuật toán khai phá dữ liệu bình thường.
- Một cách biến đổi khác phù hợp hơn là sử dụng "latent semantic analysis" (LSA) để biến đổi văn bản về dạng không thưa với số chiều thấp hơn.
- Sau phép biến đổi, chúng ta còn cần một bước scaling để giúp các văn bản với chiều dài khác nhau được xử lý đồng đều.

Khả năng biến đổi kiểu dữ liệu

Biến đổi dữ liệu chuỗi thời gian thành dữ liệu số.

Có thể sử dụng các phương pháp như “discrete wavelet transform” (DWT) hay “discrete Fourier transform” (DFT).

Biến đổi dữ liệu không gian thành dữ liệu số.

Cùng cách tiếp cận với biến đổi dữ liệu chuỗi thời gian về dữ liệu số với một số thay đổi nhỏ.

Biến đổi dữ liệu đồ thị thành dữ liệu số.

Có thể sử dụng các phương pháp như “multidimensional scaling” (MDS) hay biến đổi phổ.

Biến đổi dữ liệu chuỗi thời gian thành dữ liệu dãy rời rạc.

Có thể sử dụng phương pháp “symbolic aggregate approximation” (SAX) gồm 2 bước như sau.

- Window-based averaging.
- Value-based discretization.

Khả năng biến đổi kiểu dữ liệu

Biến đổi dữ liệu dãy rời rạc thành dữ liệu số.

- Phép biến đổi này được thực hiện với 2 bước.
 - Đổi dãy rời rạc thành một tập các chuỗi thời gian (nhị phân) mà trong đó số chuỗi thời gian bằng số kí hiệu riêng biệt trong dãy rời rạc ban đầu.
 - Nối mỗi chuỗi thời gian đó về một vector đang chiều bằng biến đổi wavelet.

Biến đổi dữ liệu bất kì thành dữ liệu đồ thị cho các ứng dụng về sự tương đồng.

- Có rất nhiều ứng dụng về sự tương đồng như bài toán gom cụm hay bài toán phát hiện ngoại lai.
- Sự tương đồng theo cặp trong dữ liệu có thể được biểu diễn bằng đồ thị tương đồng nếu như có thể định nghĩa được một hàm khoảng cách giữa các đối tượng trong dữ liệu.
- Có rất nhiều thuật toán khai phá dữ liệu có thể áp dụng cho đồ thị tương đồng.

Chuẩn bị dữ liệu

Các bước này bao gồm

- Trích xuất đặc trưng và khả năng biến đổi kiểu dữ liệu.
- **Làm sạch dữ liệu.**
- Rút gọn, chọn lọc và biến đổi dữ liệu.

Làm sạch dữ liệu

- Việc làm sạch dữ liệu rất quan trọng do quá trình thu thập dữ liệu có thể có lỗi sai hoặc thiếu sót. Một vài thí dụ như sau:
 - Một số phương tiện thu thập dữ liệu như các sensor thường có các vấn đề như hư hỏng phần cứng hoặc hết pin.
 - Thu thập dữ liệu từ các phương tiện scan có thể có lỗi do vấn đề của các kĩ thuật nhận diện kí tự từ hình ảnh.
 - Người dùng không nhập đủ hoặc nhập sai dữ liệu do lí do riêng tư.
 - Các dữ liệu được tạo thủ công có thể có lỗi trong quá trình nhập liệu.
 - Một số trường dữ liệu có thể quá tốn kém trong quá trình thu thập.

Làm sạch dữ liệu

- Các vấn đề này có thể ảnh hưởng tiêu cực đến việc khai phá dữ liệu.
- Việc làm sạch dữ liệu có một số phương diện như sau:
 - Xử lý dữ liệu bị thiếu.
 - Xử lý dữ liệu sai.
 - Scale và chuẩn hóa dữ liệu.

Làm sạch dữ liệu

Xử lý dữ liệu bị thiếu

Có **3 lớp kỹ thuật chính** cho việc xử lý dữ liệu bị thiếu.

- Loại bỏ các bản ghi có thông tin bị thiếu.
- Sử dụng các phương pháp ước lượng hoặc điền khuyết.
- Trong giai đoạn phân tích, dùng các thuật toán được thiết kế để hoạt động với dữ liệu bị thiếu.

Làm sạch dữ liệu

Xử lý dữ liệu sai và dữ liệu không đồng nhất.

Các phương pháp chính để dùng cho việc loại bỏ hoặc chỉnh sửa các dữ liệu sai hoặc không đồng nhất như sau:

- Phát hiện sự không đồng nhất.
- Sử dụng kiến thức lĩnh vực chuyên môn.
- Các phương pháp tập trung vào dữ liệu. VD: dùng các đặc tính thống kê của dữ liệu để lọc ra ngoại lai.

Làm sạch dữ liệu

Scale và chuẩn hóa dữ liệu

- Để xử lý vấn đề này, chúng ta có thể dùng các phương pháp chuẩn hóa.
- Trong nhiều tình huống thì các đặc trưng khác nhau có quy mô tham chiếu khác nhau.



Rút gọn và biến đổi dữ liệu



Rút gọn và biến đổi dữ liệu

- Khi kích thước dữ liệu nhỏ hơn, chúng ta cũng dễ áp dụng các thuật toán tinh vi và tốn kém hơn.
- Việc mất một phần thông tin do rút gọn dữ liệu có thể được bù lại bằng việc sử dụng thuật toán tinh vi hơn.




Rút gọn và biến đổi dữ liệu

Chúng ta có các loại rút gọn dữ liệu như sau:

- Lấy mẫu dữ liệu.
- Chọn lọc đặc trưng.
- Rút gọn dữ liệu với phép xoay trục.
- Rút gọn dữ liệu với phép biến đổi kiểu dữ liệu.

Lấy mẫu dữ liệu



- Lợi thế chính của lấy mẫu dữ liệu là sự đơn giản, trực quan và dễ thực hiện.
 - Cách thức lấy mẫu tùy thuộc vào ứng dụng cụ thể. Chúng ta có thể chia 2 cách lấy mẫu như sau.
 - Lấy mẫu cho dữ liệu tĩnh.
 - Lấy mẫu reservoir cho dòng dữ liệu.
- 
- 
- 

Lấy mẫu dữ liệu

Lấy mẫu cho dữ liệu tĩnh.




- Khi toàn bộ dữ liệu đã có sẵn và từ đó số điểm dữ liệu gốc được biết thì việc lấy mẫu đơn giản hơn.
- Với cách lấy mẫu không chệch, một tỉ lệ dữ liệu được định trước và giữ nguyên trong quá trình phân tích.
 - Lấy mẫu có hoàn lại.
 - Lấy mẫu không hoàn lại.

Lấy mẫu dữ liệu



Lấy mẫu cho dữ liệu tĩnh.

Ngoài ra, còn có một số loại lấy mẫu khác như sau

- Lấy mẫu chệch.
 - Lấy mẫu phân tầng.
- 
- 
- 

Lấy mẫu dữ liệu

Lấy mẫu reservoir cho dòng dữ liệu.




- Các dòng dữ liệu không có kích thước cố định mà liên tục có các điểm dữ liệu mới.
- Với cách lấy mẫu này, một mẫu với k điểm cho trước được duy trì một cách linh động từ dòng dữ liệu.
- Do kích thước rất lớn của một dòng dữ liệu, chúng ta cần các bước xử lý hiệu quả để duy trì tập mẫu k điểm với mỗi điểm dữ liệu mới từ dòng.

Lấy mẫu dữ liệu



Lấy mẫu reservoir cho dòng dữ liệu.

Với mỗi điểm dữ liệu mới, chúng ta có 2 quyết định điều khiển sau:

- Luật lấy mẫu nào để quyết định xem điểm dữ liệu mới có được cho vào mẫu hay không.
 - Luật nào để quyết định xem một điểm dữ liệu cũ trong mẫu đang có bị bỏ ra để có chỗ cho điểm dữ liệu mới.
- 
- 
- 



Lấy mẫu dữ liệu

Lấy mẫu reservoir cho dòng dữ liệu.

- Với một reservoir kích thước k điểm dữ liệu, chúng ta sẽ lấy k điểm đầu tiên trong dòng dữ liệu để khởi tạo reservoir.
- Sau đó, với điểm dữ liệu thứ n từ dòng, chúng ta có 2 quyết định điều khiển sau:
 - Cho điểm thứ n vào reservoir với xác suất k/n .
 - Nếu điểm dữ liệu mới được cho vào thì loại bỏ một trong k điểm dữ liệu cũ một cách ngẫu nhiên.



Chọn lọc đặc trưng

- Một cách khác để rút gọn dữ liệu là loại bỏ đi các đặc trưng không quan trọng.
 - Việc quyết định xem đặc trưng nào không quan trọng sẽ phụ thuộc vào ứng dụng cụ thể.
 - Có **2 loại chính** cho việc chọn lọc đặc trưng.
 - Chọn lọc đặc trưng không giám sát
 - Chọn lọc đặc trưng có giám sát
- 
- 

Giảm số chiều bằng phép xoay trục

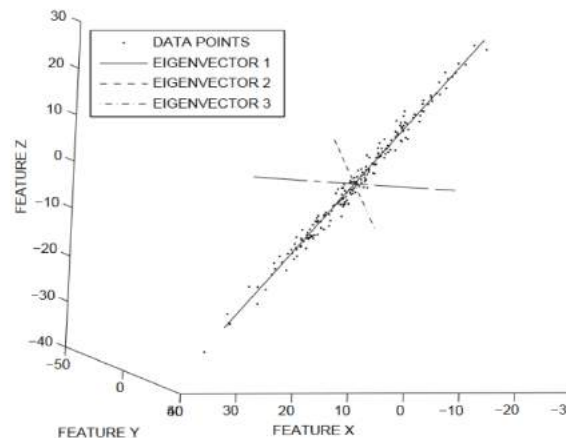
- Trong các tập dữ liệu thực tế thường có tồn tại một số đáng kể các tương quan giữa các thuộc tính khác nhau.
- Một số trường hợp, các ràng buộc giữa các thuộc tính có thể xác định được một số thuộc tính khác. VD: ngày sinh và tuổi.
- Trong hầu hết trường hợp thì các tương quan không chặt chẽ như vậy và khó có thể xác định thủ công.
- Trong hầu hết trường hợp thì các tương quan không chặt chẽ như vậy.
- Từ các ràng buộc và tương quan này, một số thông tin từ một số các chiều có thể được dùng để dự đoán thông tin của chiều khác.

Giảm số chiều bằng phép xoay trục

- Trong các tập dữ liệu thực tế thường có tồn tại một số đáng kể các tương quan giữa các thuộc tính khác nhau.
- Một số trường hợp, các ràng buộc giữa các thuộc tính có thể xác định được một số thuộc tính khác. VD: ngày sinh và tuổi.
- Trong hầu hết trường hợp thì các tương quan không chặt chẽ như vậy và khó có thể xác định thủ công.
- Trong hầu hết trường hợp thì các tương quan không chặt chẽ như vậy.
- Từ các ràng buộc và tương quan này, một số thông tin từ một số các chiều có thể được dùng để dự đoán thông tin của chiều khác.

Giảm số chiều bằng phép xoay trục

- Với dữ liệu 3 chiều như trên, việc xoay các trục theo hướng như trong hình khiến dữ liệu có thể được biểu diễn chỉ với đường thẳng 1 chiều.

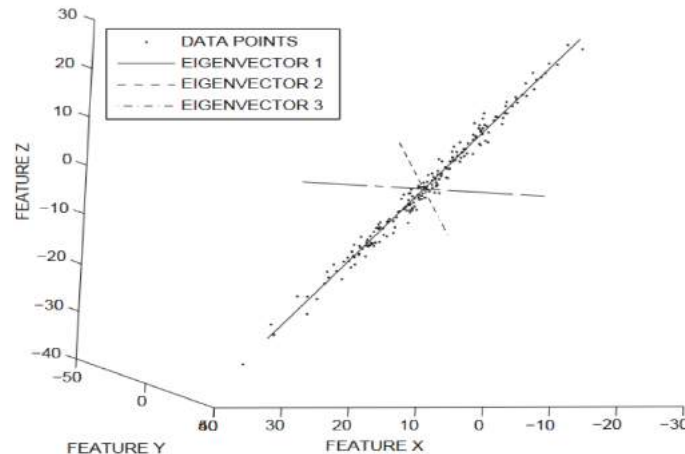


Source: Data Mining, Charu C. Aggarwal

Figure 2.2: Highly correlated data represented in a small number of dimensions in an axis system that is rotated appropriately

Giảm số chiều bằng phép xoay trục

- Để ý là 2 trục còn lại tương ứng với các chiều có phương sai thấp, do đó khi bị loại bỏ sẽ không mất quá nhiều thông tin.

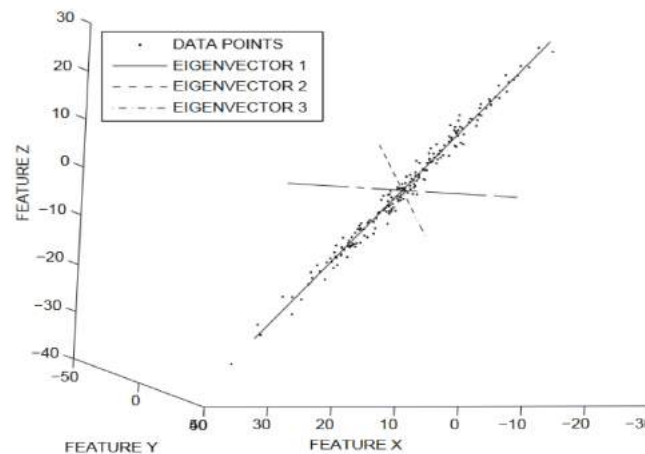


Source: Data Mining, Charu C. Aggarwal

Figure 2.2: Highly correlated data represented in a small number of dimensions in an axis system that is rotated appropriately

Giảm số chiều bằng phép xoay trục

- Từ đây chúng ta có câu hỏi rằng có cách nào tự động hóa quá trình loại bỏ này không.



Source: Data Mining, Charu C. Aggarwal

Figure 2.2: Highly correlated data represented in a small number of dimensions in an axis system that is rotated appropriately

Giảm số chiều bằng phép xoay trục

- Có 2 phương pháp cho việc này là “principal component analysis” (PCA) và “singular value decomposition” (SVD).

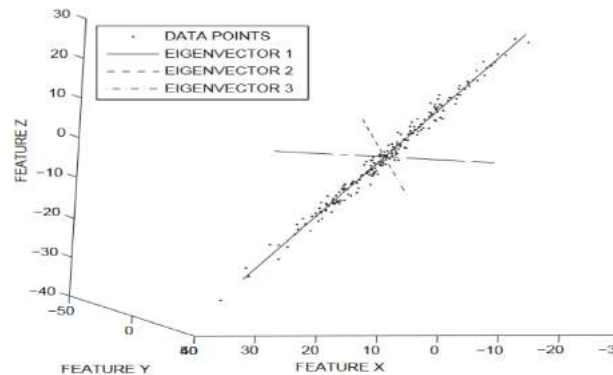


Figure 2.2: Highly correlated data represented in a small number of dimensions in an axis system that is rotated appropriately

Source: Data Mining, Charu C. Aggarwal

Giảm số chiều bằng phép xoay trục

Principal Component Analysis (PCA).

- PCA thường được dùng sau khi áp dụng mean centering cho dữ liệu, tức là mỗi điểm trong dữ liệu trừ đi trung bình dữ liệu. Khi đó, tâm của tập dữ liệu sẽ ở góc tọa độ.
- Ngoài ra, nếu trung bình dữ liệu được lưu riêng thì cũng có thể áp dụng PCA mà không cần mean centering.
- Mục tiêu của PCA là xoay dữ liệu về một hệ trục sao cho lượng phương sai lớn nhất có thể được biểu diễn bởi một số chiều nhỏ nhất.

Giảm số chiều bằng phép xoay trục




Principal Component Analysis (PCA).

- Phương sai của một tập dữ liệu theo một hướng cụ thể có thể được thể hiện thông qua ma trận phương sai của dữ liệu.
- Có thể chứng minh được ma trận phương sai là đối xứng, nửa xác định dương.
- Từ đó, ma trận này chéo hóa được và các trị riêng của ma trận phương sai biểu diễn phương sai của dữ liệu dọc theo vectors riêng tương ứng.
- Do đó các vectors riêng với trị riêng lớn sẽ thể hiện phương sai lớn hơn và được gọi là các principal component, các trục chính mới chúng ta dùng để biểu diễn dữ liệu.



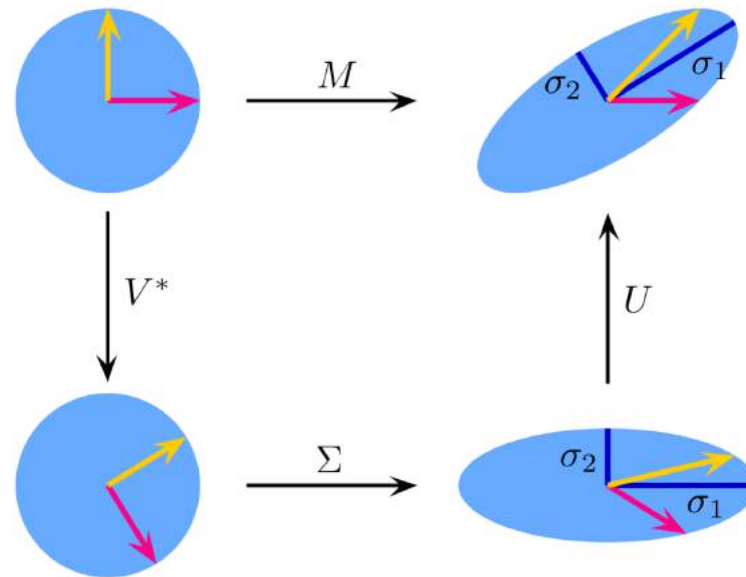
Giảm số chiều bằng phép xoay trục

Singular value decomposition (SVD)

- 
- 
- SVD có quan hệ gần với PCA.
 - SVD cho 2 bộ vector cơ sở thay vì 1 như PCA
 - SVD cho cùng vector cơ sở với PCA nếu các thuộc tính của dữ liệu có trung bình là 0.
- 

Giảm số chiều bằng phép xoay trục

- Singular value decomposition (SVD)



$$M = U \cdot \Sigma \cdot V^*$$

Source: Wikipedia - Singular value decomposition

Giảm số chiều bằng phép xoay trục

Latent Semantic Analysis (LSA)

- LSA là một ứng dụng của SVD với dữ liệu văn bản.
- Với dữ liệu văn bản, mỗi dòng của ma trận dữ liệu ứng với mỗi văn bản trong dữ liệu và chứa tần suất xuất hiện của mỗi từ của văn bản đó.
- Do đây là ma trận thưa nên trung bình của mỗi cột rất gần 0, điều này dẫn đến kết quả khá gần với PCA, mặc dù không sử dụng mean centering.
- Tính thưa của ma trận cũng dẫn đến số chiều nội tại thấp, điều này cũng dẫn đến việc giảm số chiều bằng LSA có thể rất mạnh.

Giảm số chiều bằng phép xoay trục

Ứng dụng của PCA và SVD.

- Ngoài rút giảm dữ liệu và nén dữ liệu thì PCA và SVD còn các ứng dụng khác.
 - Khử nhiễu.
 - Điền khuyết.
 - Giải hệ tuyến tính.
 - Nghịch đảo ma trận.
 - Các ứng dụng đại số ma trận khác.

Giảm số chiều bằng biến đổi kiểu dữ liệu

- Với các phương pháp này thì việc rút giảm dữ liệu đi kèm với biến đổi kiểu dữ liệu.
- Thông thường thì dữ liệu sẽ được biến đổi từ một kiểu phức tạp về một kiểu ít phức tạp hơn.
- Ở đây chúng ta sẽ tìm hiểu 2 loại phương pháp.
 - Biến đổi dữ liệu chuỗi thời gian sang dữ liệu đa chiều (Time series to multidimensional). VD: Haar wavelet transform
 - Biến đổi dữ liệu đồ thị có trọng số sang dữ liệu đa chiều (Weighted graphs to multidimensional). VD: Multidimensional scaling, spectral methods để nhúng đồ thị có trọng số vào không gian đa chiều.

Giảm số chiều bằng biến đổi
kiểu dữ liệu

Haar Wavelet Transform

- Các kĩ thuật sử dụng wavelet có rất nhiều ứng dụng đa dạng, trong đó có ứng dụng để biểu diễn dữ liệu time series theo dạng multidimensional.
- Haar wavelet là một dạng wavelet phổ biến.
- Chúng ta có thể dùng kĩ thuật wavelet để khai triển một time series thành các vector cơ sở wavelet có trọng số. Mỗi trọng số này thể hiện độ biến thiên của time series giữa 2 nửa của một khoảng thời gian.
- Trong ứng dụng giảm số chiều, các hệ số lớn (sau khi chuẩn hóa) sẽ được giữ lại.

Giảm số chiều bằng biến đổi
kiểu dữ liệu

Haar Wavelet Transform

- Demo code tham khảo về Haar wavelet.
 - https://www.numerical-tours.com/matlab/wavelet_1_haar1d/
 - https://www.numerical-tours.com/matlab/wavelet_2_haar2d/
- Ngoài ra, trên trang web trên cũng có nhiều chủ đề khác
 - <https://www.numerical-tours.com/>

Giảm số chiều bằng biến đổi
kiểu dữ liệu

Multidimensional Scaling (MDS)

- Đồ thị (graph) là một công cụ mạnh mẽ để biến diễn quan hệ giữa các đối tượng.
- Trong các ứng dụng khai phá dữ liệu thì các đối tượng có thể có kiểu dữ liệu rất phức tạp và không thuần nhất.

Giảm số chiều bằng biến đổi
kiểu dữ liệu

Multidimensional Scaling (MDS)

- Đồ thị (graph) là một công cụ mạnh mẽ để biến diễn quan hệ giữa các đối tượng.
- Trong các ứng dụng khai phá dữ liệu thì các đối tượng có thể có kiểu dữ liệu rất phức tạp và không thuần nhất.

Giảm số chiều bằng biến đổi
kiểu dữ liệu

Multidimensional Scaling (MDS)

- Tuy nhiên, tùy vào mục đích ứng dụng mà chúng ta có thể định nghĩa được khoảng cách từng cặp đối tượng.
- Với các đối tượng thế này thì MDS là một cách tự nhiên để hỗ trợ hình dung khoảng cách giữa các đối tượng.
- Cụ thể hơn là từ các khoảng cách theo cặp giữa các đối tượng tạo mà nhúng các đối tượng này vào 1 không gian đa chiều (*với số chiều được chọn*) và từ đó có thể sử dụng các thuật toán cho dữ liệu không gian đa chiều.

Giảm số chiều bằng biến đổi
kiểu dữ liệu

Spectral Transformation and Embedding of Graphs.

- Nếu như MDS được thiết kế để bảo toàn khoảng cách toàn cục thì các phương pháp phổ (spectral methods) được thiết kế để bảo toàn khoảng cách địa phương.
- Các phương pháp phổ hoạt động với các đồ thị tương đồng (similarity graph) thay vì khoảng cách.
- Nếu có dữ liệu khoảng cách thì chúng ta có thể dùng các hàm kernel như heat kernel để chuyển khoảng cách về độ tương đồng.